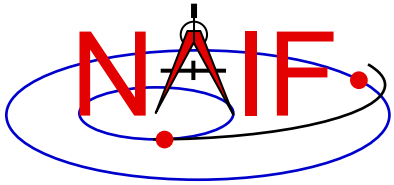


Navigation and Ancillary Information Facility

Leapseconds and Spacecraft Clock Kernels

LSK and SCLK

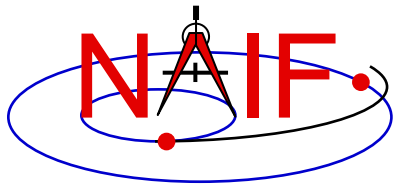
October 2022



Topics

Navigation and Ancillary Information Facility

- **Kernels Supporting Time Conversions**
 - LSK
 - SCLK
- **Forms of SCLK Time Within SPICE**
- **Backup**



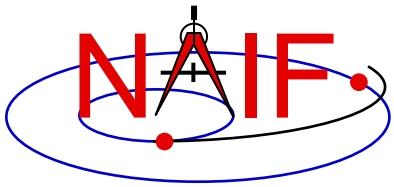
SPICE Time Conversion Kernels

Navigation and Ancillary Information Facility

In most cases one or two kernel files are needed to perform conversions between supported time systems.

- **LSK - The leapseconds kernel is used in conversions between ephemeris time (ET/TDB) and Coordinated Universal Time (UTC).**
- **SCLK - The spacecraft clock kernel is used in conversions between spacecraft clock time (SCLK) and ephemeris time (ET/TDB).**
 - **It's possible there could be two or more clocks associated with a given spacecraft.**

Ephemeris Time, ET and Barycentric Dynamical Time, TDB, are the same

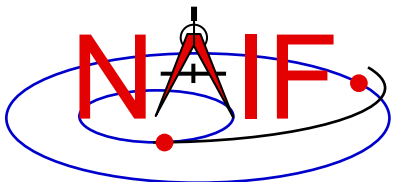


The Leapseconds Kernel (LSK)

Navigation and Ancillary Information Facility

The leapseconds kernel contains a tabulation of all the leapseconds that have occurred, plus additional terms.

- **Used in ET \leftrightarrow UTC and in ET \leftrightarrow SCLK conversions.**
 - Subroutines using LSK: STR2ET, TIMOUT, ET2UTC, etc.
 - Utility programs using LSK: *spkmerge*, *chronos*, *spacit*, etc.
- **Use FURNISH to load it.**
- **NAIF updates the LSK when a new leap second is announced by the International Earth Rotation Service (IERS).**
 - The latest LSK file is always available from the NAIF server.
 - » The latest is LSK always the best one to use.
 - Announcement of each new LSK is typically made months in advance using the “spice_announce” system.
 - » http://naif.jpl.nasa.gov/mailman/listinfo/spice_announce
 - New LSKs take effect **ONLY** on January 1st and July 1st



LSK File Example

Navigation and Ancillary Information Facility

KPL/LSK

. . . <comments> . . .

\begindata

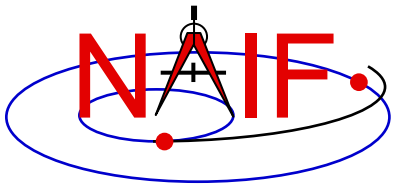
```
DELTET/DELTA_T_A      = 32.184
DELTET/K               = 1.657D-3
DELTET/EB              = 1.671D-2
DELTET/M               = ( 6.239996D0  1.99096871D-7 )
```

```
DELTET/DELTA_AT       = ( 10,   @1972-JAN-1
                          11,   @1972-JUL-1
                          12,   @1973-JAN-1
                          13,   @1974-JAN-1
                          14,   @1975-JAN-1
```

. . . <more leapsecond records> . . .

```
35,   @2012-JUL-1
36,   @2015-JUL-1
37,   @2017-JAN-1 )
```

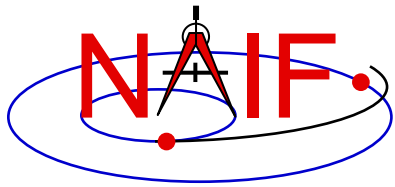
\begintext



Out of Date LSKs

Navigation and Ancillary Information Facility

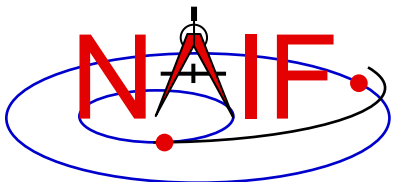
- An out-of-date leapseconds kernel can be used successfully for conversions that occur at epochs **prior** to the epoch of the first missing leapsecond.
 - Any conversions of epochs occurring after the epoch of a missing leapsecond will introduce inaccuracies in multiples of one second per missed leapsecond.
- Using the latest leapseconds kernel to perform conversions at epochs more than six months ahead of the last leapsecond listed may result in an error if, later on, a new leapsecond is declared for a time prior to the epochs you processed.



The Spacecraft Clock Kernel (SCLK)

Navigation and Ancillary Information Facility

- **The spacecraft clock kernel is required by Toolkit utilities and routines that utilize SCLK time.**
 - For example, the SPICE CK subsystem makes heavy use of spacecraft clock time.
- **Use FURNISH to load it.**
- **Ensure you have the correct version of the SCLK file for your spacecraft since this kernel may be updated rather frequently.**
 - SCLK files are usually maintained on a flight project server.
 - » For JPL operated missions they can always be found on the NAIF server as well.
 - When using a CK, “correct SCLK” means compatible with that CK.
 - » For reconstructed CKs, this is most likely the latest version of the SCLK.
 - » For “predict” CKs, this is probably the SCLK kernel used when the CK was produced.



SCLK File Example

Navigation and Ancillary Information Facility

KPL/SCLK

. . . <comments> . . .

\begindata

SCLK_KERNEL_ID = (@2009-12-07/18:03:04.00)

SCLK_DATA_TYPE_74 = (1)

SCLK01_TIME_SYSTEM_74 = (2)

SCLK01_N_FIELDS_74 = (2)

SCLK01_MODULI_74 = (4294967296 256)

SCLK01_OFFSETS_74 = (0 0)

SCLK01_OUTPUT_DELIM_74 = (1)

SCLK_PARTITION_START_74 = (0.00000000000000E+00

. . . <more partition start records> . . .
2.4179319500800E+11)

SCLK_PARTITION_END_74 = (2.0692822929300E+11

. . . <more partition end records> . . .
1.0995116277750E+12)

SCLK01_COEFFICIENTS_74 = (

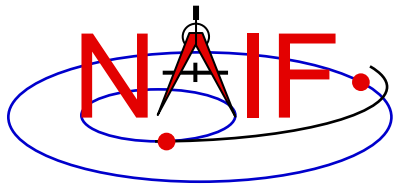
0.00000000000000E+00 -6.3119514881600E+08 1.00000000000000E+00

1.2098765056000E+10 -5.8393434781600E+08 1.00000000000000E+00

. . . <more coefficient records> . . .

2.4179319365000E+11 3.1330950356800E+08 9.9999997500000E-01)

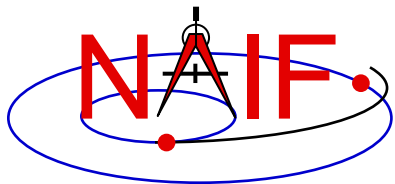
\begintext



Forms of SCLK Time Within SPICE

Navigation and Ancillary Information Facility

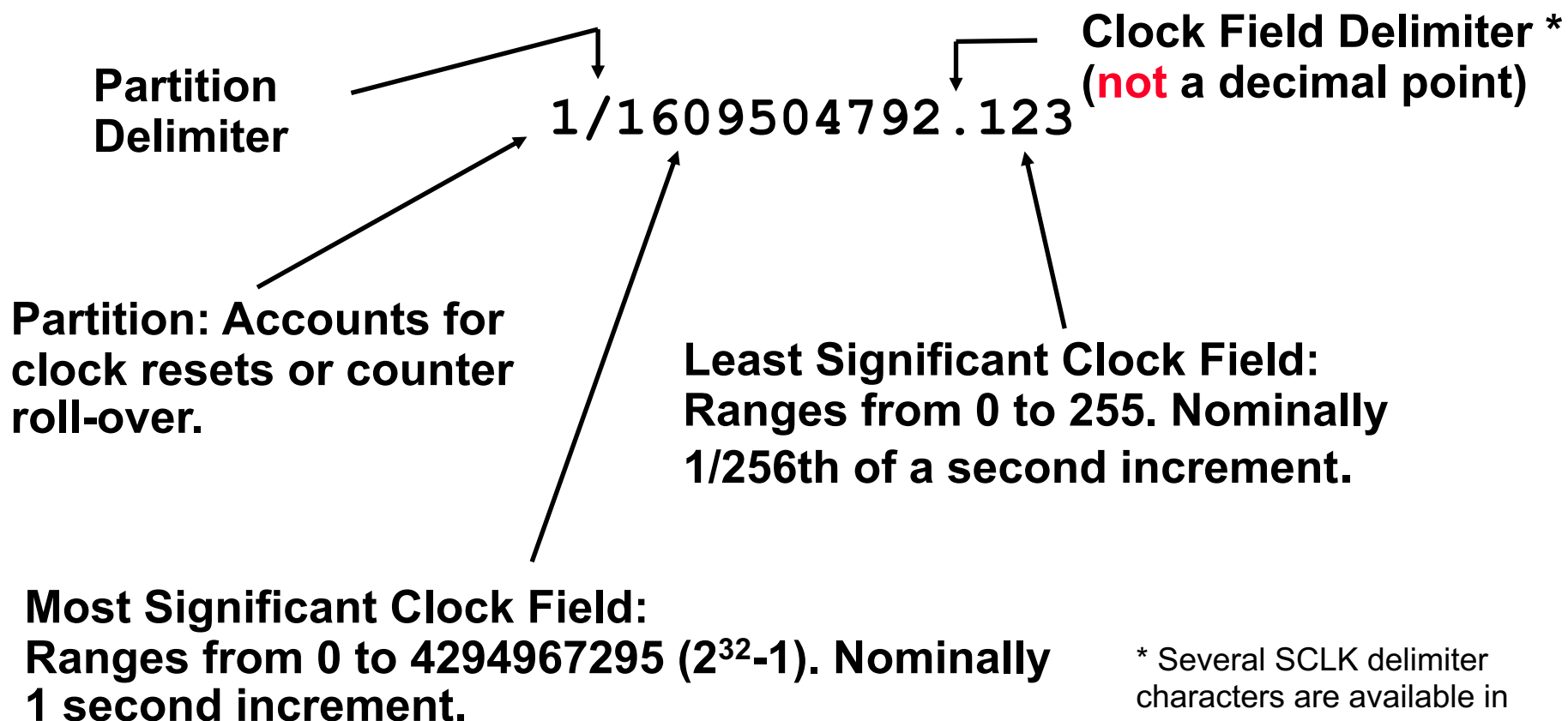
- **SCLK time in SPICE is represented in two different ways:**
 - a character string
 - a double precision number called “ticks”
- **A SCLK character string is composed of one or more cascading integer numbers – similar to a digital clock.**
 - This form is derived from clock values represented by sets of bits or bytes found in downlinked telemetry.
- **A SCLK value encoded as a double precision number (called “ticks”) is used within SPICE because it’s easy to convert this to other time systems, such as ephemeris time.**



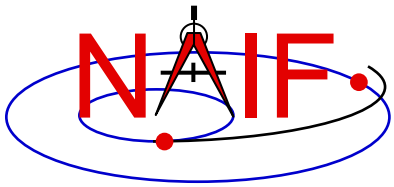
Sample SCLK String

Navigation and Ancillary Information Facility

The Cassini orbiter SCLK time string consists of three fields separated by delimiters.



* Several SCLK delimiter characters are available in SPICE. See "SCLK Required Reading" for details.



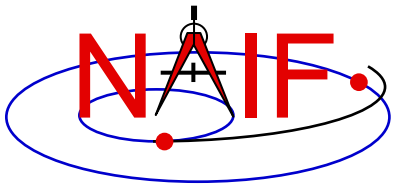
What is a Partition?

Navigation and Ancillary Information Facility

①/1609504792.123

The portion of the SCLK string circled above indicates the partition to which the remaining portion of the string is related.

- **A partition is a NAIF-created construct to handle spacecraft clock rollovers or resets.**
- **SCLK strings not having a partition number are treated as belonging to the first partition in which they occur.**

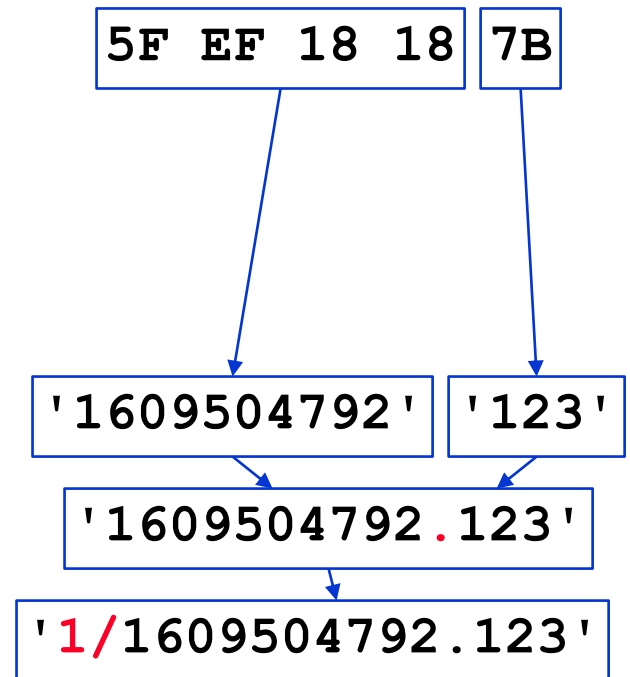


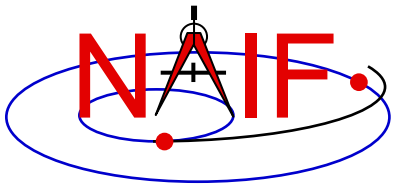
Constructing a SCLK String - 1

Navigation and Ancillary Information Facility

Usually SCLK tags in raw telemetry are represented by **sets of bits or bytes**. Such tags must be converted to SCLK strings used in SPICE. This is an example of how it is done for the sample CASSINI SCLK string from previous slides.

- Start with a **5-byte** CASSINI TLM SCLK
 - The first four bytes are an unsigned integer representing seconds
 - The last byte is an unsigned byte representing fractional seconds (as a count of 1/256 second ticks)
- Convert integer seconds and integer fractional second ticks to two strings
- Concatenate strings together using a recognized delimiter ('.', ':', etc)
- Add the partition number and delimiter
 - Optional; for most modern missions it may be omitted (not so for Chandrayaan-1 and MESSENGER)



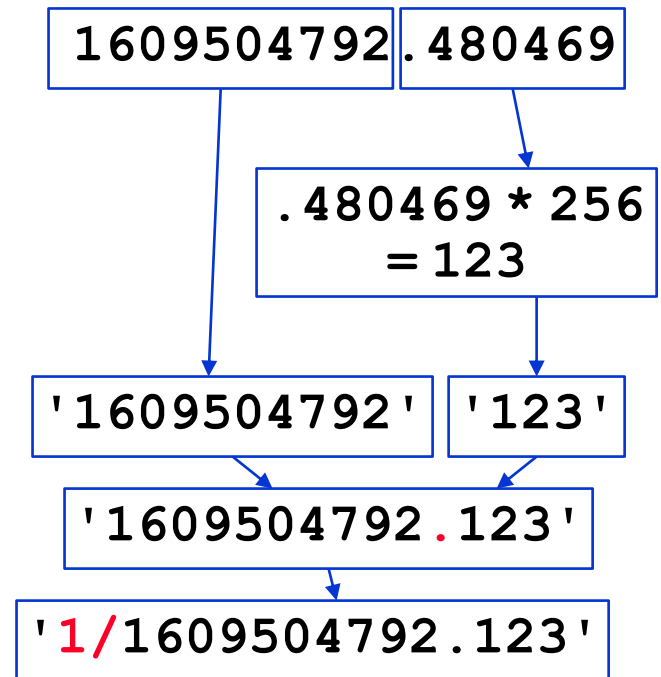


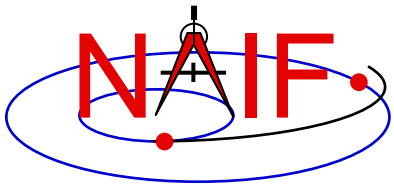
Constructing a SCLK String - 2

Navigation and Ancillary Information Facility

On modern missions SCLK tags in raw telemetry are also often represented as **DP numbers**. Such tags must also be converted to SCLK strings used in SPICE. This is an example of how it is done for the same sample CASSINI SCLK string.

- Start with a **DP number** CASSINI TLM SCLK
- Convert decimal fraction to integer count of ticks
- Convert integer seconds and integer fractional second ticks to two strings
- Concatenate strings together using a recognized delimiter ('.', ':', etc)
- Add the partition number and delimiter
 - Optional; for most modern missions it may be omitted (not so for Chandrayaan-1 and MESSENGER)



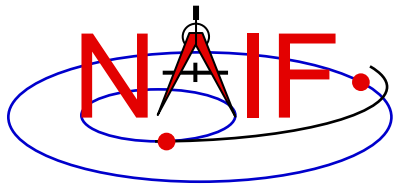


Encoded SCLK (Ticks)

Navigation and Ancillary Information Facility

The representation of SCLK time in the SPICE system is a double precision encoding of a SCLK string.

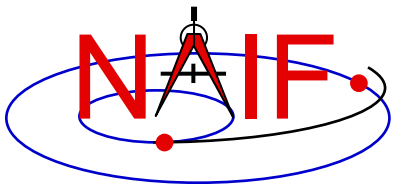
- **Encoded spacecraft clock values represent “ticks since spacecraft clock start.”**
 - The time corresponding to tick “0” is mission dependent and does not necessarily relate to launch time. It is often an arbitrary epoch occurring before launch.
- **A tick is the smallest increment of time that a spacecraft clock measures.**
 - For example, in the case of the Cassini orbiter this is nominally 1/256th of a second.
- **Encoded SCLK increases continuously independent of leapseconds, clock resets, and clock rollovers.**



Additional Info on LSK and SCLK

Navigation and Ancillary Information Facility

- **For more information about LSK, SCLK, and time conversions, look at the following documents**
 - Time Required Reading
 - SCLK Required Reading
 - Time tutorial (at the end it has a nice graphic depicting time APIs)
 - Most Useful SPICELIB Routines
 - headers for the routines mentioned in this tutorial
 - CHRONOS User's Guide
 - Porting_kernels tutorial
- **Related documents**
 - Kernel Required Reading
 - CK Required Reading



SCLK Interface Routines

Navigation and Ancillary Information Facility

Convert SCLK times using the following routines

SCS2E (SC, SCLKCH, ET)

(SCLK String \Rightarrow ET)

SCE2S (SC, ET, SCLKCH)

(ET \Rightarrow SCLK String)

SCT2E (SC, SCLKDP, ET)

(Encoded SCLK \Rightarrow ET)

SCE2C¹ (SC, ET, SCLKDP)

(ET \Rightarrow Continuous Encoded SCLK)

SCE2T (SC, ET, SCLKDP)

(ET \Rightarrow Discrete Encoded SCLK)

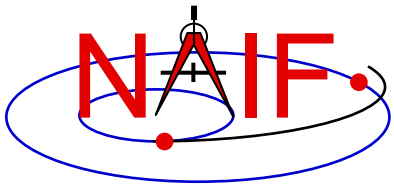
SCENCD (SC, SCLKCH, SCLKDP)

(Encode SCLK string to ticks)

SCDECD (SC, SCLKDP, SCLKCH)

(Decode SCLK ticks to string)

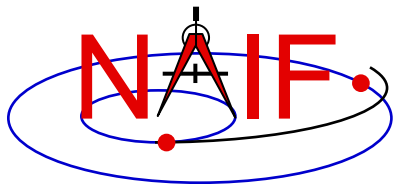
¹ Use SCE2C (not SCE2T) for C-kernel data access.



Backup

Navigation and Ancillary Information Facility

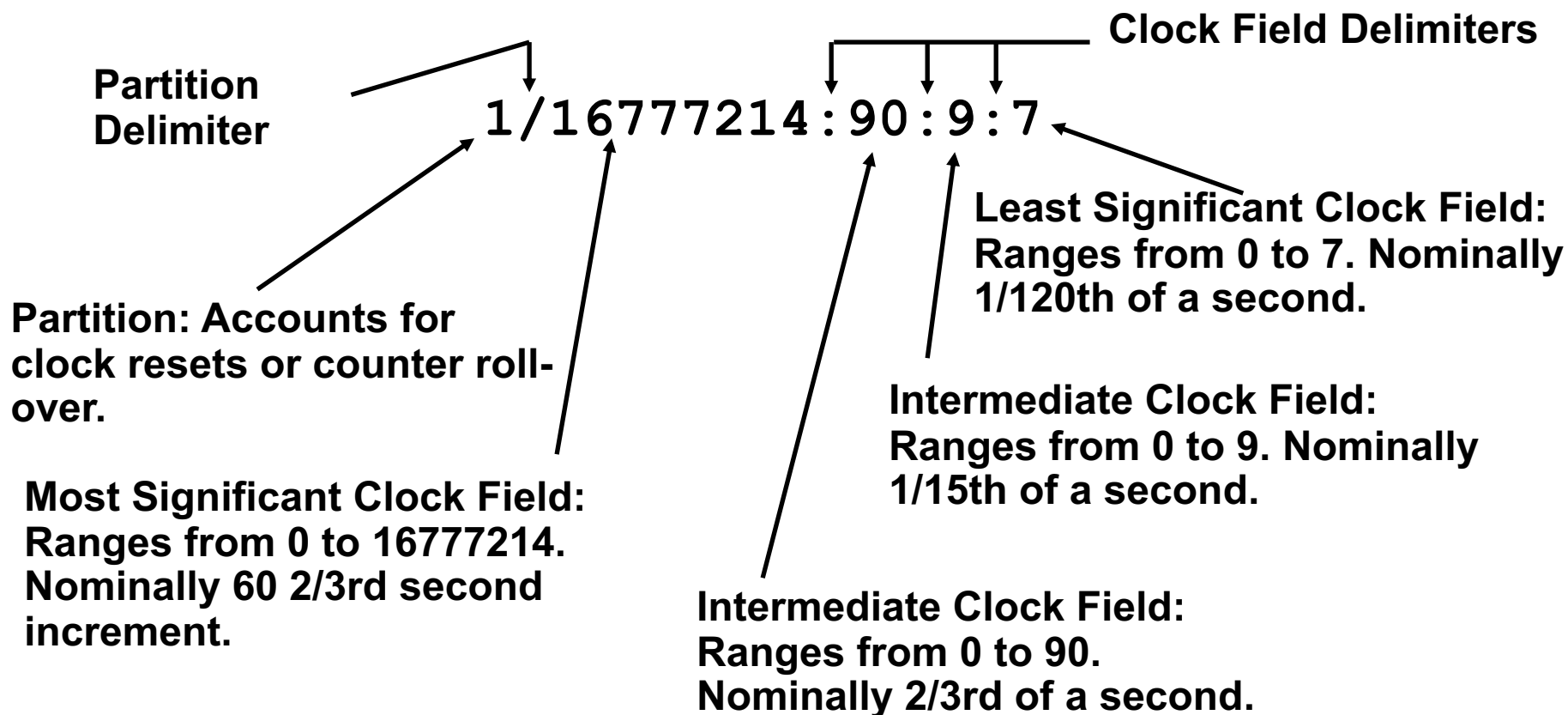
- **Examples of SCLK strings**
- **Dates of "recent" leapseconds**

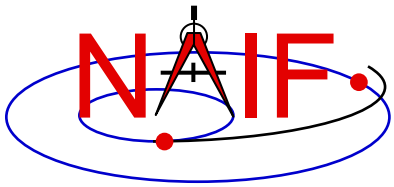


Sample Galileo SCLK String

Navigation and Ancillary Information Facility

The Galileo spacecraft SCLK time string consists of five fields separated by delimiters.





More Sample SCLK Strings

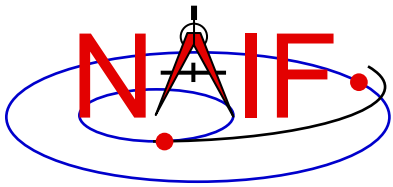
Navigation and Ancillary Information Facility

The following are examples of SCLK strings* from missions using SPICE.

- **Cassini**
1/1334314108.134
- **DS1**
1/67532406.010
- **Galileo**
1/16777214:90:9:7
- **Genesis**
1/666230496.204
- **MGS**
1/655931592.103
- **MPF**
1/559627908.058
- **Mariner 9**
1/11542909
- **Mars Odyssey**
1/687231994.091
- **NEAR**
1/40409721942
- **Stardust**
1/697451990.042
- **Viking 1&2**
1/32233616
- **Voyager 1&2**
1/05812:00:001
- **Mars Express**
1/0090979196.29713
- **Venus Express**
1/0033264000.50826
- **Rosetta**
1/0101519975.65186

* When clock strings are used as arguments in modules they must be contained in quotes:

- Single quotes for Fortran
- Double quotes for C
- Single quotes for IDL and MATLAB



Dates of New Leapseconds⁺

Navigation and Ancillary Information Facility

1972-JAN-1
1972-JUL-1
1973-JAN-1
1974-JAN-1
1975-JAN-1
1976-JAN-1
1977-JAN-1
1978-JAN-1
1979-JAN-1
1980-JAN-1
1981-JUL-1
1982-JUL-1
1983-JUL-1
1985-JUL-1
1988-JAN-1
1990-JAN-1
1991-JAN-1
1992-JUL-1
1993-JUL-1
1994-JUL-1
1996-JAN-1
1997-JUL-1
1999-JAN-1
2006-JAN-1
2009-JAN-1
2012-JUL-1
2015-JUL-1
2017-JAN-1

- **New leapseconds become effective only on January 1st or July 1st.**
- **NAIF announces every new leapsecond using the "spice_announce" Mailman system**
- **NAIF provides a new leapseconds kernels* (LSK) several months before the new leapsecond becomes effective**

⁺ Current as of January 1, 2020

* NAIF provides both Linux/OSX and Windows versions of the LSK